

AngularJS პლატფორმასთან მუშაობა Node.js ვებ-სერვერის მეშვეობით

გიორგი კენჭოშვილი
საქართველოს ტექნიკური უნივერსიტეტი

რეზიუმე

განიხილება AngularJS - ის წარმოშობის ისტორია და მისი უპირატესობები. ნაჩვენებია AngularJS ბიბლიოთეკის, Bootstrap CSS სტილების ჩამოტვირთვა და სამუშაო გარემოს გამართვა. შემოთავაზებულია ვებ-სერვერისა და ტესტირების სისტემის ინსტალაციის ნიმუშები. საილუსტრაციო მაგალითის სახით განხორციელებულია მარტივი SPA-აპლიკაციის ჩატვირთვა ბრაუზერში.

საკვანძო სიტყვები: Unit ტესტი. End-To-End ტესტი. Npm. Node.js. Karma. Jasmine Framework.

1. შესავალი

AngularJS წარმოადგენს ღია კოდის (Open Source) მქონე JavaScript ტექნოლოგიას, რომელიც შეიმუშავა კომპანია Google-მა. პირველად იგი გაუღერდა 2009 წელს Misko Hevery-ის მიერ, როგორც JS მონაცემებთან სამუშაო სერვისის კომპონენტი. სერვისი განთავსებული იყო დომენზე GetAngular.com და გააჩნდა რეგისტრირებული მომხმარებლები, თუმცა საბოლოოდ გადაწყდა ბიზნეს იდეის უარყოფა და გამოუშვეს AngularJS ბიბლიოთეკა ღია კოდით[3]. დღეს-დღეობით, კომპანია Google აქტიურად განაგრძობს ამ ტექნოლოგიის განვითარებასა და მხარდაჭერას. AngularJS წარმოადგენს კლიენტის მხარის დაპროგრამების ტექნოლოგიას. იგი გამდიდრებულია სერვერული მხარის ტექნოლოგიებიდან აღებული საუკეთესო ფუნქციონალით და გამოიყენება ვებ-ბრაუზერში ჩატვირთული HTML-დოკუმენტების გასამდიდრებლად. ამგვარად, AngularJS ტექნოლოგია აიოლებს მდიდარი სამომხმარებლო ინტერფეისის მქონე, ინტერაქტიული ვებ-აპლიკაციების შემუშავების პროცესს. AngularJS ვებ-აპლიკაციები ეფუძნება Model-View-Controller (MVC) არქიტექტურულ სტანდარტს, რაც საშუალებას იძლევა შევიმუშაოთ აპლიკაციები, რომლებიც არიან:

- განვრცობადი (Extendible): პროგრამისთვის იოლია გაერკვეს კომპლექსური AngularJS ვებ-აპლიკაციის ლოგიკასა და სტრუქტურაში, დაამატოს ახალი ფუნქციონალი[1].
- მხარდაჭერა (Maintenance): AngularJS აპლიკაციები იოლად ექვემდებარება Debug პროცესს, შემოწმებასა და შემდგომ მხარდაჭერას.
- ტესტირებადი (Testable): AngularJS აპლიკაციები მორგებულია ტესტირების პროცესზე: Unit და End-To-End ტესტების შემუშავება გაიოლებულია. ეს კი იმას ნიშნავს, რომ ავტომატურ რეჟიმში შეგვიძლია გაუშვათ ტესტები, ვიპოვოთ და აღმოვფხვრავთ ხარვეზები, მანამ, სანამ საბოლოო მომხმარებელი გადააწყდება მათ.
- სტანდარტიზებული (Standardized): AngularJS აგებულია თანამედროვე ვებ-სტანდარტების გათვალისწინებით და სარგებლობს უახლესი ტექნოლოგიების ფუნქციონალით (როგორცაა HTML 5 API), პოპულარული ხელსაწყოებითა და არქიტექტურით.

შემდეგში განიხილება AngularJS -ისთვის ვებ-სერვერის და ტესტირების სისტემის ინსტალაცია.

2. ვებ-სერვერის ინსტალაცია

კლიენტის მხარის დაპროგრამების ხელსაწყოთა უმრავლესობა დაწერილია JavaScript ენაზე და ეშვება Node.js გარემოში[1].

Node.js წარმოადგენს Google Chrome ვებ-ბროუზერში ჩაშენებულ JavaScript ძრავს, თუმცა ასევე გათვალისწინებულია ბროუზერის გარეთ მუშაობის რეჟიმშიც. გადავიდეთ მისამართზე: <https://nodejs.org> და დავაყენოთ Node.js პაკეტი (გათვალისწინებულია Windows, Linux და Mac OS ოპერაციული სისტემების შესაბამისი საინსტალაციო ვერსიები). იმისათვის, რომ დავრწმუნდეთ Node.js-ის ინსტალაციის კორექტულად შესრულებაში, გავუშვათ Command Line რედაქტორი. აკრიფოთ `node` და გავუშვათ ბრძანება (`enter` ღილაკზე დაჭერით). დაველოდოთ, სანამ არ შეიცვლება `prompt` კურსორი და გავუშვათ შემდეგი კოდი (კოდი უნდა აკრიფოთ ერთ ხაზზე):

```
function testNode() {return "Node is working"}; testNode();
```

შედეგად Node.js გამოთვლის ჩვენს მიერ აკრიფილ JavaScript კოდს და გამოიტანს შედეგს:

```
'Node is working'
```

მრავალი გზა არსებობს Node.js ვებ-სერვერად გამოსაყენებისთვის დასაკონფიგურირებლად. ჩვენ ავირჩიეთ ყველაზე მარტივ და საიმედო მეთოდი, რაც გულისხმობს add-on მოდულების ინსტალაციას Node.js-ის ლოკალურ საინსტალაციო დირექტორიაში (ჩვენს შემთხვევაში: `C:\Program Files\nodejs`). დაპროგრამების პროცესისთვის მარტივი ვებ-სერვერის გამართვა საკმარისი იქნება. ამიტომ, სერვერს შევქმნის მიზნით გამოვიყენებთ Node.js-ის მოდულებს სახელად `connect` და `serve-static`.

გავუშვათ Command Line რედაქტორი და დავდგეთ Node.js-ის საინსტალაციო დირექტორიაზე. ამ მიზნით `cmd`-ში უნდა აკრიფოთ `cd` [საინსტალაციო დირექტორიის მისამართი], რაც ჩვენს შემთხვევაში იქნება:

```
cd C:/Program Files/Nodejs
```

და გავუშვათ ბრძანებები:

```
npm install connect და npm install serve-static.
```

NPM წარმოადგენს Node-ის პაკეტების ინსტალატორს. იგი ჩამოტვირთავს ყველა საჭირო ფაილს, რაც `connect` და `serve-connect` მოდულებს ჭირდებათ. ამის შემდეგ შევქმნათ ახალი ფაილი სახელად `server.js` (კვლავ ვიმყოფებით Nodejs საინსტალაციო დირექტორიაში (ნახ.1).

```
var connect = require('connect');
var serveStatic = require('serve-static');

var app = connect();

app.use(serveStatic("../angularjs"));
app.listen(5000);
```

ნახ.1. Serve.js ფაილის კონტენტი

ეს მარტივი ვებ-სერვერი საკმარისია დეველოპმენტის პროცესისთვის. იგი ყურს უდებს პორტ 5000-ზე შემოსულ მოთხოვნებს და ემსახურება ფაილებს, რომლებიც იმყოფებიან `angularjs` დირექტორიაში. ხოლო `angularjs` დირექტორია იმყოფება იმავე დონეზე, რომელზეც `nodejs` საინსტალაციო დირექტორია.

3. ტესტირების სისტემის ინსტალაცია

AngularJS ტექნოლოგიის დიდი ერთ-ერთი უპირატესობა არის Unit Testing ტესტირების სისტემასთან მაღალი შესაბამისობის დონე[2]. ეს საშუალებას გვაძლევს დეველოპმენტის პროცესშივე ავტომატურად შევამოწმოთ კოდის სისწორე. ლექციებში ტესტების გასაშვებად გამოვიყენებთ კარმა Test Runner ტესტირების გარემოს და Jasmine ტესტირების Framework-ს. ორივე მათგანი ფართოდ არის გავრცელებული და ადვილად მოხმარებადი. Node.js საინსტალაციო დირექტორიდან გავეუშვათ შემდეგი ბრძანება:

```
npm install -g karma
```

NPM-ი გადმოწერს ყველა იმ ფაილს, რაც Karma-ს ესაჭიროება. შემდგომი კონფიგურირება აღარ არის საჭირო. არმა ტესტირების გარემო Unit Test ტესტებს უშვებს Node.js სერვერზე, ხოლო თავად ტესტი შეგვიძლია შევიშუშოთ Jasmine, Unit ან სხვა ბიბლიოთეკების გამოყენებით. ამ ეტაპზე Karma-ს მუშაობის პრინციპებს არ ჩავუღრმავდებით.

4. AngularJS დირექტორიის შექმნა

შემდეგ ეტაპზე უნდა შევქმნათ დირექტორია, საიდანაც მოვემსახურებით AngularJS აპლიკაციებს დეველოპმენტის პროცესში. ეს საშუალებას გვაძლევს დაპროგრამების პროცესშივე შევამოწმოთ სამუშაოს პროგრესირება და ორგანიზება გავეუკეთოთ ფაილებს. შევქმნათ საქალაქო სახელწოდებით angular.js. ახალი საქალაქე იმავე დონეზე უნდა იმყოფებოდეს, რომელზეც შევქმენით nodejs საინსტალაციო საქალაქე.

5. AngularJS ბიბლიოთეკის ჩამოტვირთვა

გადმოვიწეროთ AngularJS ტექნოლოგიის უახლესი დასტაბილურებული ვერსია მისამართიდან: <http://angularjs.org>. მთავარ გვერდზე დავაჭიროთ Download და დავრწმუნდეთ, რომ Stable და uncompressed ოფციები არის ჩართული (როგორც მე-2 ნახაზზეა ნაჩვენები). ასევე შესაძლებელია არა-სტაბილური (Prerelease) ან მინიმიზირებული (minified) ვერსიის გადმოწერა. სალექციო კურსში გამოვიყენებთ AngularJS-ის არა-კომპრესირებულ ვერსიას. შევინახოთ angular.js ფაილი ახლად შექმნილ angularjs დირექტორიაში[1]. AngularJS ბიბლიოთეკის უახლესი ვერსია არის 1.4.8, რომელიც მალე შეიძლება ახალი ვერსიით ჩანაცვლდეს, თუმცა სტაბილური რელიზის შემთხვევაში AngularJS ფუნქციონალი არ ირღვევა და უფრო ახალი ვერსიის დაყენება არ გამოიწვევს პრობლემებს.



ნახ.2. AngularJS ბიბლიოთეკის გადმოწერა

ფაილი	აღწერა	
angular-touch.js	Touchscreen Event-ის მხარდაჭერა	
angular-animate.js	ანიმაციები კონტენტის ცვლილებისას	
angular-route.js	URL მარშრუტიზაცია	
angular-sanitize.js	სახიფათო კონტენტის არიდება	
angular-locale-fr-fr.js	ლოკალიზაციის დეტალები, ენა-ფრანგული. ლოკალიზაციის ფაილების ერთ-ერთი წარმომადგენელი, i18n დირექტორიაში.	

6. Bootstrap სტილების ბიბლიოთეკა

კონტენტის სტილებით გასამდიდრებლად გამოვიყენებთ Bootstrap CSS ბიბლიოთეკას. Bootstrap არ წარმოადგენს AngularJS ტექნოლოგიასთან სამუშაო სავალდებულო ნაწილს, არც რაიმე კავშირი არსებობს ამ ორ პაკეტს შორის, მაგრამ Bootstrap-ს გააჩნია კარგი CSS სტილების ნაკრები. იგი საშუალებას გვაძლევს იოლად შევიმუშაოთ HTML-დოკუმენტები და ყოველ ჯერზე არ ვწეროთ საკუთარი CSS-სტილები [1].

გადავიდეთ გზავნილზე: <http://getbootstrap.com> და დავაჭიროთ Download Bootstrap ლილაკს. მივიღებთ არქივს, რომელიც შეიცავს JavaScript და CSS ფაილებს. მათგან მხოლოდ ორი ფაილი დაგვჭირდება: დავაკოპირებთ შემდეგი ფაილები angularjs საქალაქში (ამ ეტაპზე საქალაქში იქნება მხოლოდ angular.js ფაილი).

- bootstrap.css
- bootstrap-theme.css

Bootstrap შედგება CSS და JavaScript ფაილებისგან. სალექციო კურსში გამოვიყენებთ მხოლოდ CSS ფაილებს, ხოლო Javascript ფაილები არ დაგვჭირდება.

სანამ სასწავლო მაგალითებზე გადავალთ, შევამოწმოთ, რომ ყველაფერი დაინსტალირებულია და მუშაობს. შევქმნათ HTML ფაილი სახელად test.html და დავამატოთ angularjs დირექტორიაში.

```

<!DOCTYPE html>

<html ng-app>
<head>
  <title>First Test</title>
  <script src="angular.js"></script>
  <link href="bootstrap.css" rel="stylesheet" />
  <link href="bootstrap-theme.css" rel="stylesheet" />
</head>
<body>
  <div class="btn btn-default">{{AngularJS}}</div>
  <div class="btn btn-success">Bootstrap</div>
</body>

```

ნახ.3. AngularJS და Bootstrap სტილების შემოწმება test.html ფაილში

კოდში ზოგიერთი ნაწილი შეიძლება უცხო იყოს, მაგალითად ng-app ატრიბუტი html ელემენტზე და {{AngularJS}} გამოსახულება body ელემენტზე - წარმოადგენს AngularJS ტექნოლოგიის ნაწილს, ხოლო btn, btn-default და btn-success კლასები მოდიან Bootstrap ბიბლიოთეკიდან. ამ ეტაპზე ჩვენი მიზანია დავრწმუნდეთ სამუშაო გარემოს გამართულობაში.

7. სატესტო HTML-დოკუმენტის ჩატვირთვა

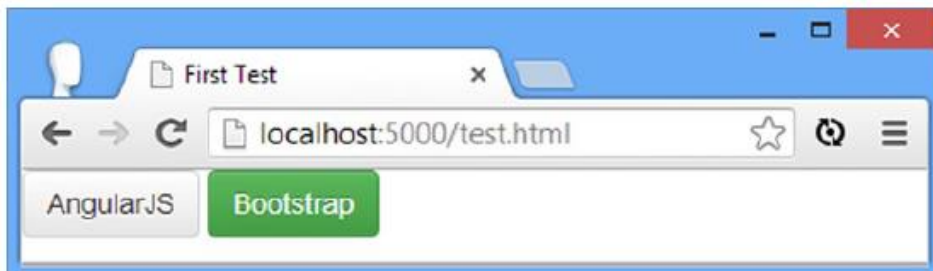
ვებ-სერვერის გაშვება ხდება Command Prompt-ში (cmd). დავდგეთ Nodejs საინსტალაციო დირექტორიაზე (ჩვენს შემთხვევაში უნდა ავკრიფოთ cd C:\ როგრამ Files\nodejs) და გავუშვათ ბრძანება: node server.js

ეს ჩატვირთავს წინა ნაწილში ანგულარჯს დირექტორიაში შექმნილ server.js ფაილს და პორტზე ნომრით 5000 ჩაირთვება HTTP მოთხოვნებზე ყურის გდების რეჟიმი.

გავხსნათ Chrome ბროუზერი და მისამართის ველში ავკრიფოთ:

<http://localhost:5000/test.html>.

შედეგად ბროუზერში ჩაიტვირთება ახლად-შექმნილი HTML-დოკუმენტი:



ნახ.4. სამუშაო გარემოს შემოწმება

მე-5 ნახაზზე კი ნაჩვენებია შემთხვევა, როდესაც არც AngularJS და არც Bootstrap არ მუშაობს კორექტულად. ყურადღება მივაქციოთ ორმაგი ფიგურული ფრჩხილის სიმბოლოებს {{ და }} და არც კონტენტი არის წარმოდგენილი დოკუმენტის სახით (რაც უნდა შეესრულებინა Bootstrap ბიბლიოთეკას). ამ შემთხვევაში უნდა გადავამოწმოთ ვებ-სერვერის კონფიგურაცია და angularjs საქაღალდეში მოთავსებული ფაილები.



ნახ.5. გაუმართავი სამუშაო გარემოს შემთხვევა

8. დასკვნა

ერთგვერდიანი აპლიკაციები (SPA) სრულიად ახალი, განსხვავებული მიდგომა თანამედროვე ვებ-აპლიკაციების დაპროგრამების მეთოდოლოგიაში. AngularJS აპლიკაციებში გაიოლებულია ტესტების შემუშავების პროცესი, რაც Dependency injection - ის დამსახურებაა. იოლად ექვემდებარება Debug პროცესს, სარგებლობს უახლესი ტექნოლოგიების ფუნქციონალით, არქიტექტურით, ხელსაწყოთა ნაკრებით და ა.შ..

სწორედ ზემოთ ჩამოთვლილი ფაქტორების გათვალისწინებით, AngularJS ტექნოლოგია სარგებლობს მეტი პოპულარულობით სხვა წინამორბედ ტექნოლოგიებთან შედარებით.

ლიტერატურა:

1. Freeman A. (2014). Pro AngularJS (Expert's Voice in Web Development). 1st ed., Apress, ch. 1, pp. 3-15.
2. Karpov V., Netto D. (2015). Professional AngularJS, ch. 9, pp. 275-315.
3. კენჭოშვილი გ. (2016). ერთგვერდიანი JAVASCRIPT - პლატფორმების ANGULARJS და EMBERJS განხილვა და შედარება. შრომები: „მართვის ავტომატიზებული სისტემები No 2(20), 2016“. სტუ. თბ., გვ.130-135.

ANGULAR.JS PLATFORM TO WORK WITH NODE.JS WEB SERVER THROUGH

Kentchoshvili Giorgi

Georgian Technical University

Summary

Considered AngularJs - production history and its benefits. Displaying AngularJs library, Bootstrap CSS style sheets, and download a working environment on. Proposed web - server system installation and testing of samples. An illustrative example is embodied in its SPA - Application to load the browser.

РАБОТА С ПЛАТФОРМОЙ ANGULARJS ПРИ ПОМОЩИ ВЕБ-СЕРВИСА NODE.JS

Кенчошвили Г.

Грузинский Технический Университет

Резюме

В статье рассматривается история создания Angular.Js, его преимущества. Показан пример загрузки библиотеки Angular.Js и стилей Bootstrap CSS, а также настройки рабочего пространства. Автором представлены примеры инсталляции веб-сервера и системы тестирования. В качестве иллюстрации рассмотрена загрузка в браузер простейшего SPA-приложения.